Quebrando a Cifra

A Sildávia e a Volgônia estão em guerra. O Serviço Secreto Sildaviano conseguiu interceptar mensagens do alto comando do exército volgonês, e eles acreditam que essas mensagens contêm informações que podem garantir uma vitória sildaviana no conflito. Infelizmente, as mensagens estão criptografadas. O governo sildaviano empregou o mais famoso matemático do país, Alan Knutjikstra, para resolver o problema, e, após vários meses trabalhando, ele conseguiu quebrar o código.

As mensagens transmitidas são, na verdade, programas que devem ser executados em um computador especial. Esse computador possui uma memória de 10^6 posições, numeradas de zero a 10^6-1, sendo que todas possuem inicialmente valor zero. Ele possui apenas três instruções:

- ADD A B I, que adiciona o valor I a todas as posições de memória entre A e B, inclusive. I pode estar entre -1000 e 1000, inclusive. A e B são números válidos de posições de memória, e A é sempre menor ou igual a B.
- PRINT P, onde P é um número inteiro entre 1 e 10^6-1, que imprime um caractere na tela. O caractere impresso é determinado pela diferença entre o valor da posição P e o valor da posição P-1 na memória. Mais especificamente, se a diferença for negativa, então é impresso um espaço em branco. Se a diferença for positiva, então considere o resto da divisão dela por 26, e use esse valor como um offset em relação à letra A. Por exemplo, se a diferença é 106, então o resto da divisão de 106 por 26 é 2, o que corresponde à letra C.
- HALT, que termina o programa.

Uma mensagem é transmitida como uma sequência de inteiros, cujo tamanho varia de 1 até 4.000.000. A mensagem é uma sequência de instruções, onde a instrução ADD é representada pelo inteiro 18, a instrução PRINT é representada pelo inteiro 42 e a instrução HALT é representada pelo inteiro 0. Cada instrução é seguida por seus argumentos, se houverem.

Por exemplo, considere a sequência:

18 2 2 2 18 1 3 5 42 1 42 2 18 0 0 1 0

Ela corresponde ao seguinte programa:

```
ADD 2 2 2
ADD 1 3 5
PRINT 1
PRINT 2
ADD 0 0 1
HALT
```

Que imprime a mensagem "FC".

Você vai receber como entrada uma mensagem interceptada. Decodifique-a.

Entrada

A entrada é composta de várias mensagens. Cada mensagem é dada em 2 linhas. A primeira linha contém um inteiro N (0 < N ≤ 4.000.000), que é o número de inteiros na mensagem. A segunda linha contém N inteiros, todos os quais são menores que 2^(30) em valor absoluto. A entrada termina com N = 0, que não deve ser processado.

Você pode supor que todas as mensagens na entrada são válidas, i.e., correspondem a um programa válido que possui pelo menos uma instrução HALT. Note que ainda podem haver mais dados após a instrução HALT, e que esses dados não necessariamente correspondem a instruções válidas. Dado que a execução do programa termina com o HALT, você deve ignorar tudo que vier após ele.

Saída

Para cada mensagem na entrada, imprima uma linha na saída padrão no formato MENSAGEM: [x], onde x é o texto decodificado da mensagem.

Observações

A mensagem decodificada pode ser vazia.

Exemplos

Entrada:
7
18 1 1 1 42 1 0
1
0
0

Saída:
MENSAGEM: [B]
MENSAGEM: []